

«

»

• • , • •

- :

122 « , » ,  
« »

: , . - . .  
 , . - . .

. , . .

( 10 21.06.2018 . )  
 ( 10 30 2018 . )

, . .  
 , . .  
 - :

[ ] : . . 122  
 « , » , «  
 » / . . , . . ; .  
 . - (1 , 4.64 ). - : .  
 , 2018. - 52 .

« 6.050101 « , - » ,  
 , ,  
 « , » ,  
 .

© . . , . .  
 © . , 2018

.....	4
.....	6
.....	20
.....	24
.....	47
,	48
.....	51
.....	52

« -  
 » ( - 2016 .)  
 ( ) « »  
 122 « , »  
 « ».

( ).  
 —  
 122 « , »  
 « »  
 ( ) « », ,  
 ,  
 .  
 - :  
 — :  
 - « » 6.050101 « ,  
 »:  
 (7/ );  
 (9/ ); - (8/ );  
 - 122 « , »  
 « »:  
 (1/ );

—  
 .  
 ,  
 ,  
 .  
 .  
 ,  
 ,  
 ,  
 ,  
 .  
 :  
 1. ( ):  
 - ,

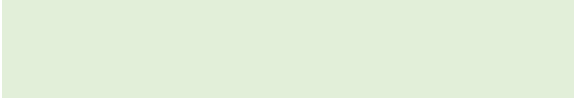
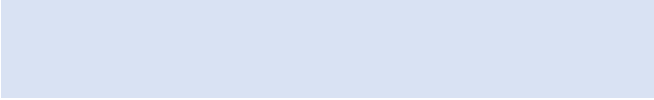
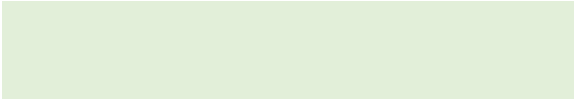
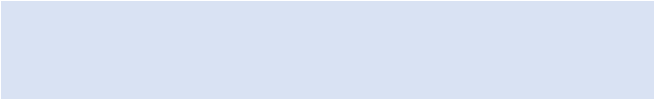
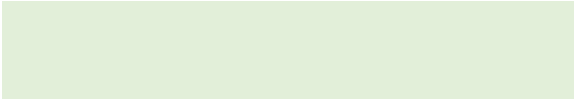
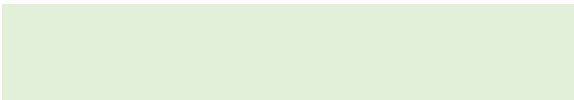
- ;
- 
- 2. :
- ;
- ,
- 3. : - ,
- 150 (5
- ECTS). ( ) .
- : 28
- , 26 ( ,
- ), 96
- .

, .  
 , - /  
 ( ).  
 : Agile ( )  
 ): Scrum Kanban, Jira ( )  
 ).  
 ,  
 ( ) — ,  
 . ,  
 ,  
 . ( :  
*Agile software development, agile-* )  
 , DSDM, Scrum Kanban.  
 , ,  
 . / ( 2-3 ),  
 .  
 , -  
 : , , ,  
 . , / .  
 / , ,  
 .  
 Agile- (face-to-face).  
 , ( ,  
 , - ).  
 , ,  
 .  
 — . ,  
 ,

Agile Manifesto

*Agile Manifesto.*

( . 1),



. 1. Agile Manifesto

- :
- - , ,
  - 
  - 
  -
- :
- 1.
  - 2.
  - 3.
  - 4.

5.

,

,

.

6.

—

(face-

to-face).

7.

—

.

8.

,

.

9.

.

10.

—

.

11.

,

.

12.

,

.

,

.

:

-

Agile-

,

.

,

.

-

,

,

,

,

.

:

.

-

,

Agile

.

,

.

,

.

.

,

Agile

Manifesto:

- *Agile Modelling* —

,

,

( . 2).

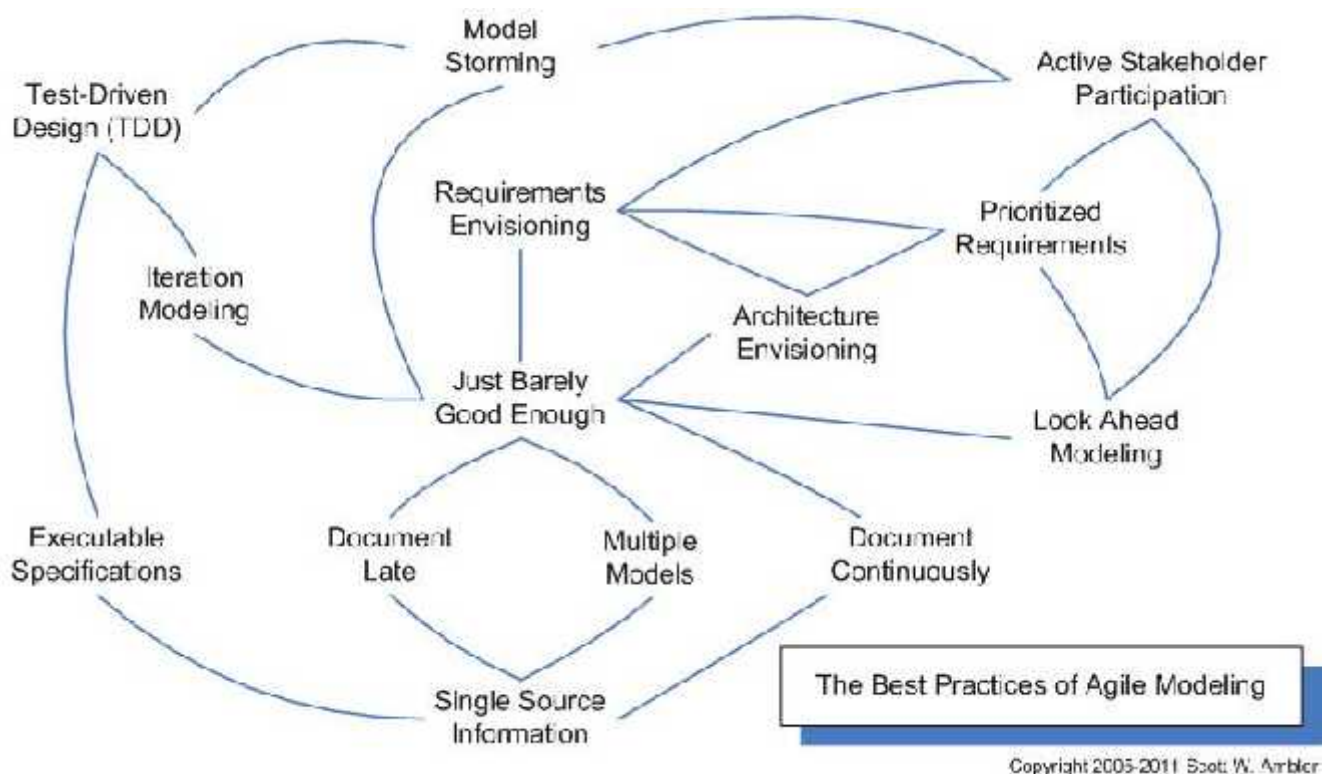
,

UML.

—

,





. 2.

## Agile Manifesto

- **Agile Unified Process** – ( / )
- **Agile Data Method** – ,
- **DSDM** – (Rapid Application Development, RAD).
- **Feature driven development (FDD)** – , « ».
- **Getting Real** – ,

- *OpenUP* –

-

:

,

,

.

.

.

,

.

,

.

- *Scrum* –

,

.

.

-

.

Scrum ( « »).

,

.

.

,

.

**Scrum** –

,

,

,

( . 3).

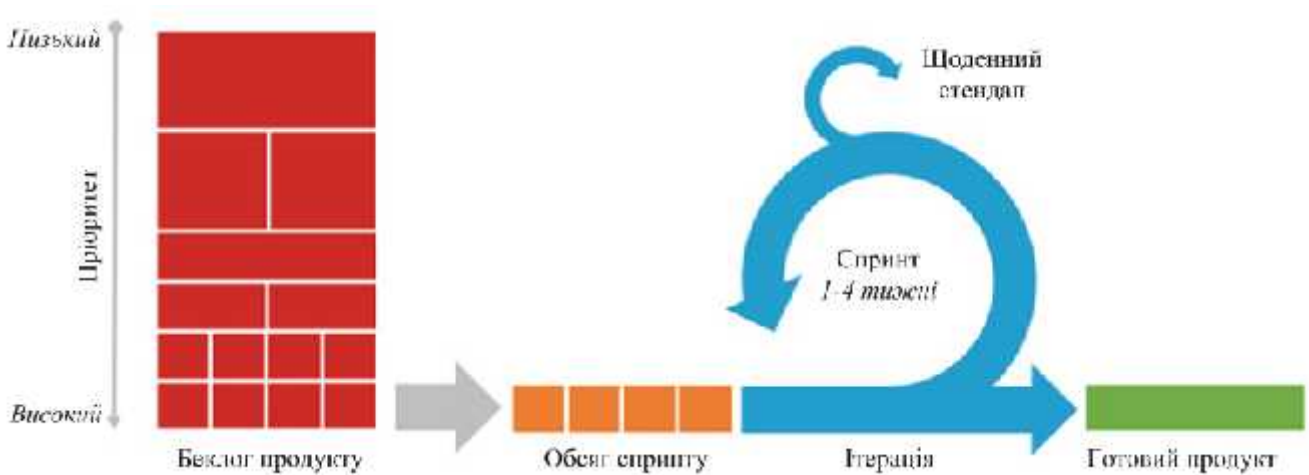
.

( )

( ).

,

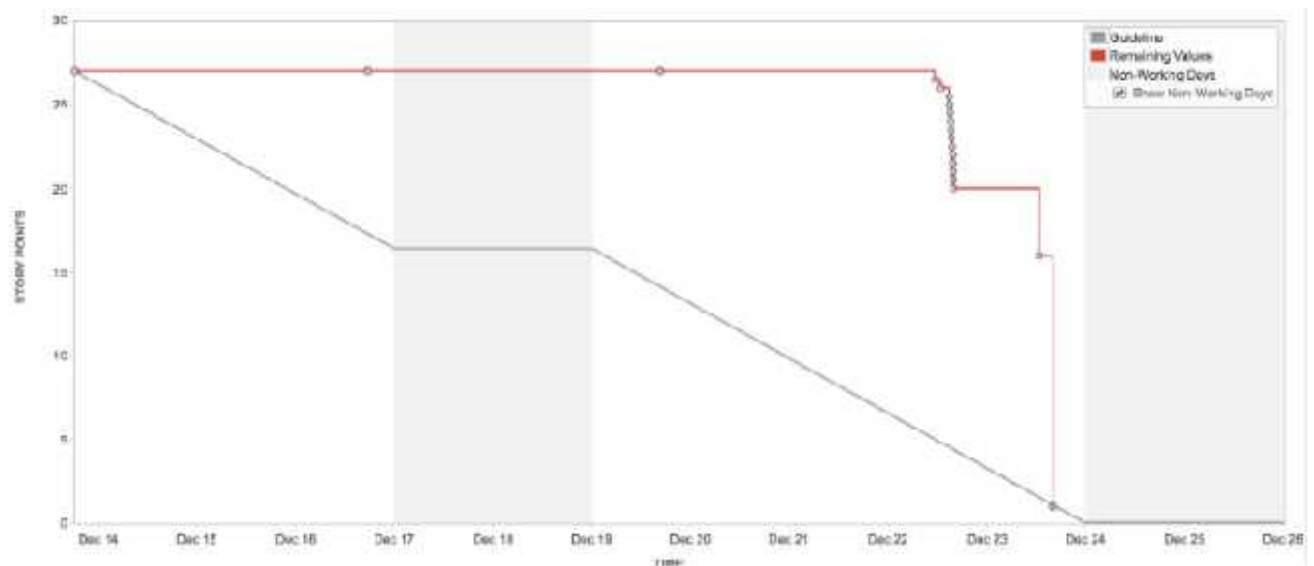
.



. 3.

Scrum





#### 4. Burndown chart

1)

2)

(

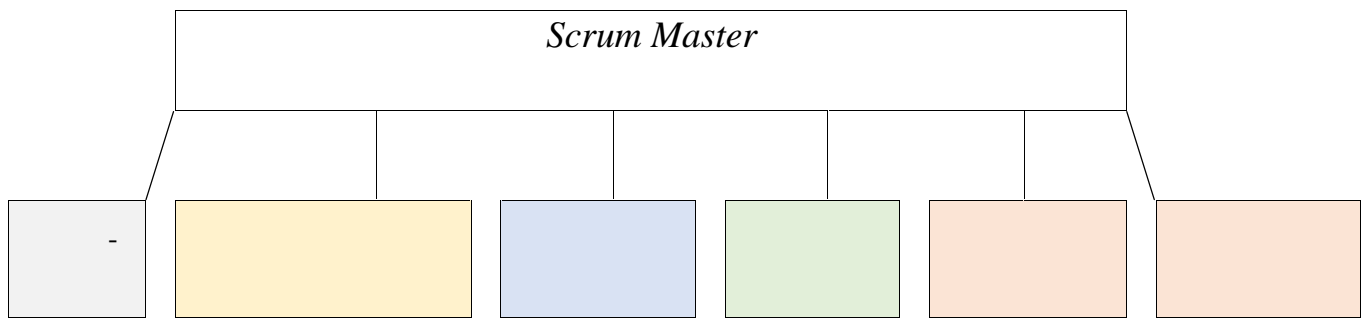
Scrum

*Scrum Master* (

).

Scrum,

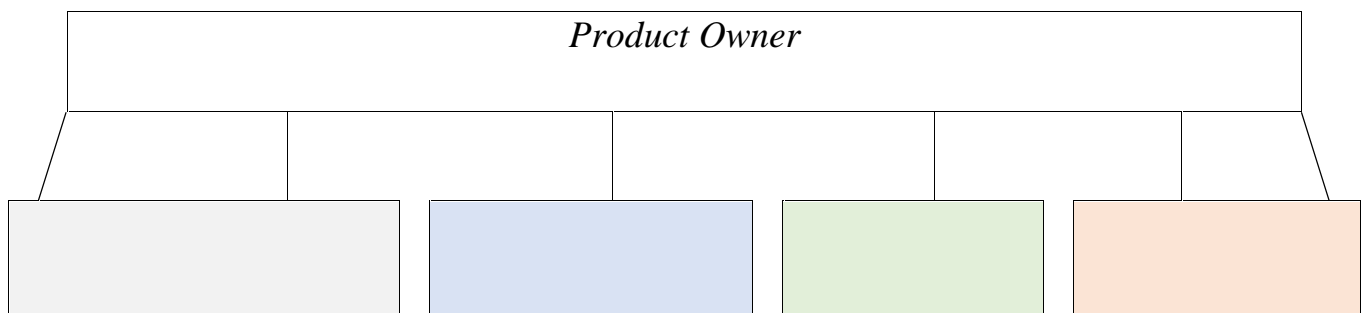
( . 4).



. 4. ,

*Product Owner* ( ). *Product manager* , . *Product Owner* -

Product Owner (0.5):



. 5. ,

Product Owner

Scrum

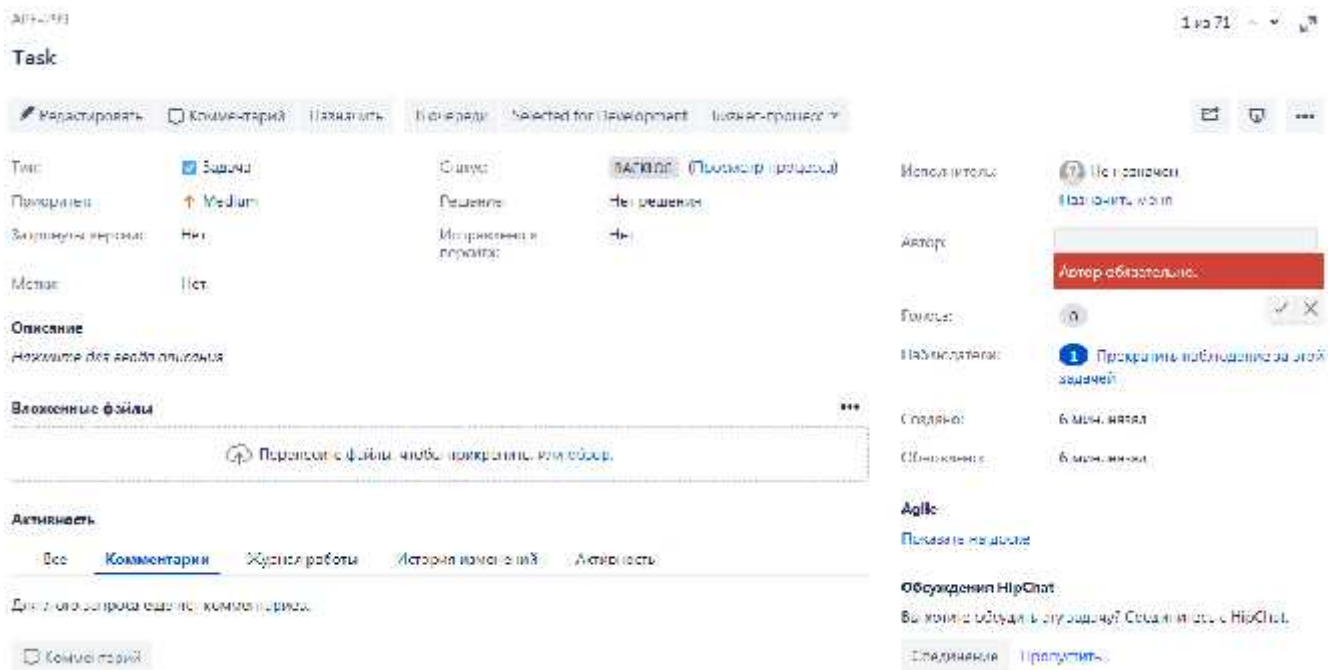
Scrum

Team

Team



« » ( Kanban  
: ;  
— — ;  
— — ;  
— ;  
— « ».  
: ,  
Kanban:  
( 5 );  
( Kanban  
Backlog- );  
Jira. **Jira** —  
**Jira** —  
Java EE (**Java Enterprise Edition**,  
Java).  
**Jira** —  
. JIRA  
Atlassian. JIRA 2002  
(ticket or  
issue).  
( )  
( . 7).



## 7. Jira

workflow

( , ).

/ , , .

;

;

( , )

.

### Jira

SOAP, XML-RPC REST.

Jira

: Eclipse IntelliJ IDEA. Jira,

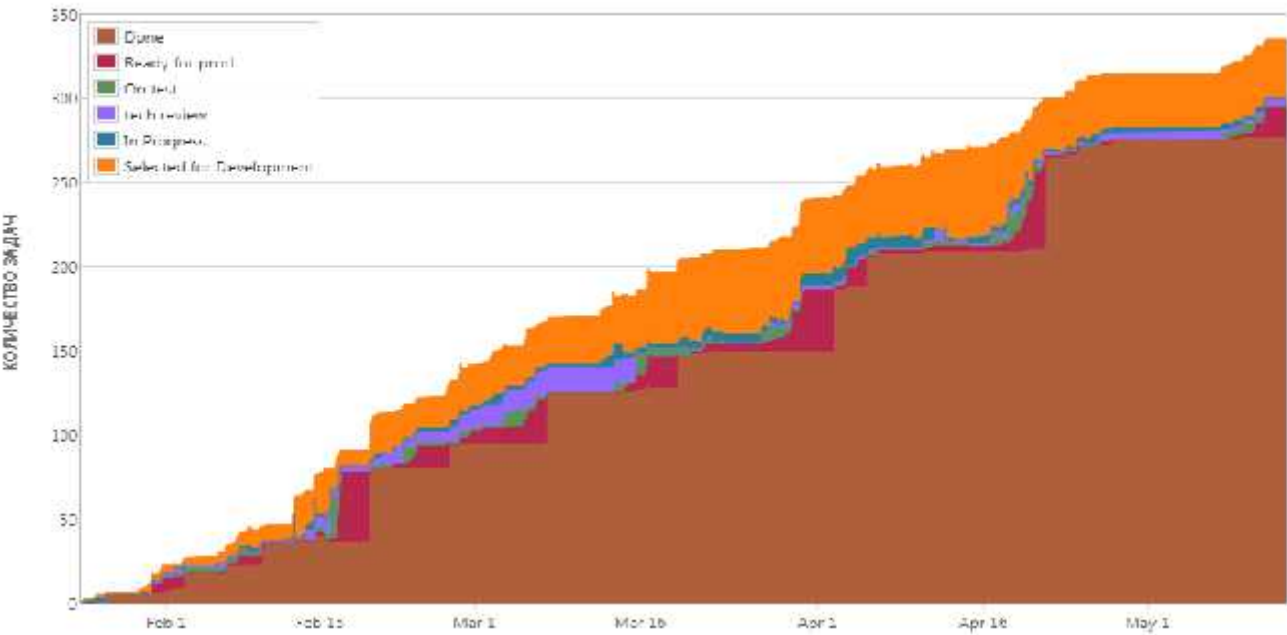
JIRA,



1. ( . 8).

«Select for Development»  
«Done»  
«» —

24/АВГ/18 8:13/МАР/18 (ВСЕ ВРЕМЯ) УТОЧНИТЬ ОТВЕТ



. 8.

2. ( . 9).



. 9.

**Jira** –

это система управления задачами и проблемами, которая позволяет команде работать над проектами, отслеживать прогресс и управлять ресурсами. Jira используется для управления задачами, проблемами, проектами и другими элементами работы. Jira позволяет создавать задачи, устанавливать приоритеты, назначать ответственных и отслеживать их выполнение. Jira также предоставляет инструменты для анализа данных, такие как диаграммы Ганта, диаграммы PERT и другие. Jira является популярным инструментом для управления проектами и проблемами в компаниях.

- Trello.
- Rally

- Asana
- Pivotal Tracker
- Bugzilla
- Basecamp
- Redmine
- VersionOne.
- GitHub Issues
- Axosoft
- .







;

—

;

—

9

•

•

---

,

,

,

•  
,

---

,

,

•

1 - (2/ .01)

1.1. \_\_\_\_\_  
 \_\_\_\_\_ (2/ .01.01)

1.2. \_\_\_\_\_  
 \_\_\_\_\_ (2/ .01.02)

1.3. \_\_\_\_\_, \_\_\_\_\_  
 \_\_\_\_\_ (2/ .01.03)

**1.4** \_\_\_\_\_ (2/ .02.01)

2 (2/ .02)

**2.1** \_\_\_\_\_ (2/ .02.01)

**2.2** \_\_\_\_\_ (2/ .02.03)

**2.3** \_\_\_\_\_ - \_\_\_\_\_ (2/ .02.03)



ISO9000:

ISO12207:

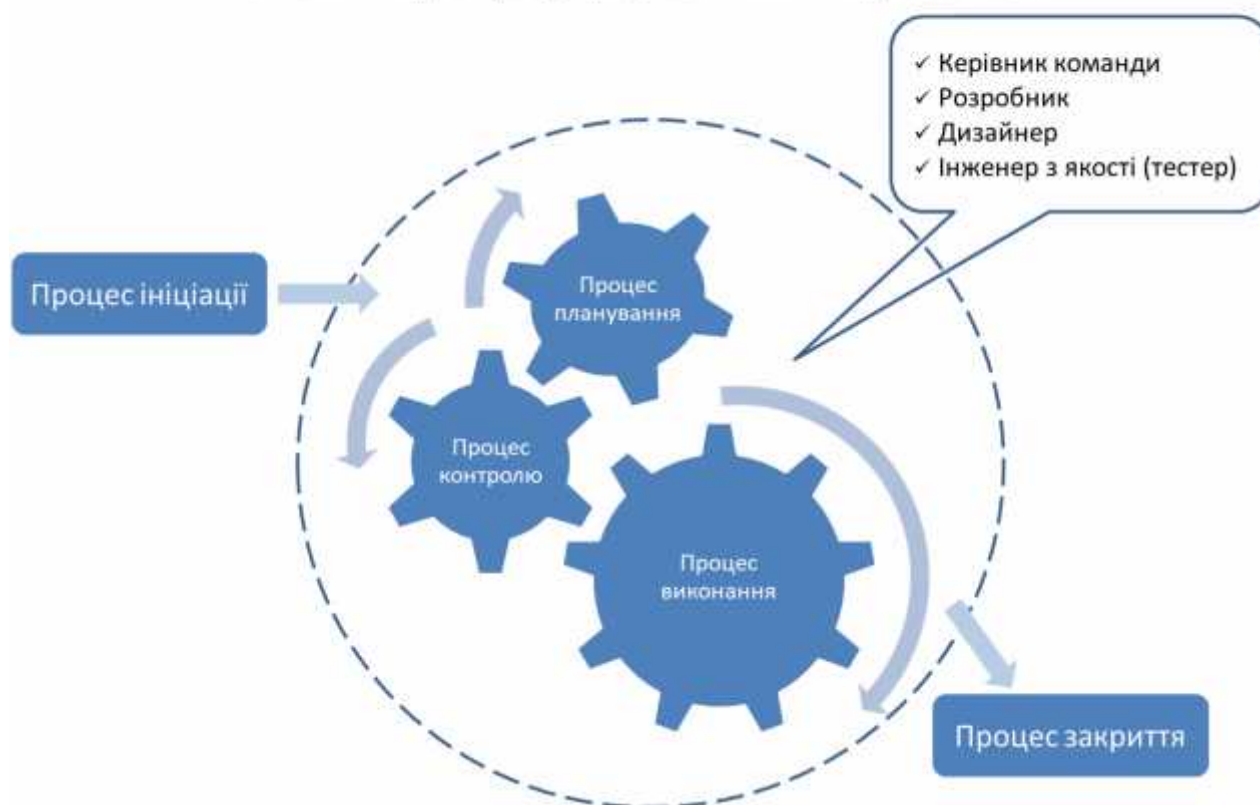
ISO 15504:

2.4 \_\_\_\_\_ (2/ .02.04)

## Проектний менеджмент



## Схема процесу управління проектом



**Проект** – це обмежена часовими рамками діяльність, що має визначений початок та кінець, зазвичай обмежений датою, але також може обмежуватися фінансуванням або досягненням результатів, яка здійснюється для реалізації унікальних цілей та завдань, зазвичай, щоб призвести до вигідних змін або створення доданої вартості.



Проект має бути гарно структурований для ефективної роботи. Мета структуризації/декомпозиції зводиться до наступних складових проекту:

- фаза життєвого циклу проекту, етапу, задачі;
- організаційна структура виконавців проекту;
- структура розподілу відповідальності;
- загальні системні функції, які виконуються на всіх фазах реалізації проекту.

**Фаза проекту** – визначений обмежений проміжок часу виконання проекту, що раціонально відокремлений по відношенню до інших проміжків.

**Життєвий цикл** – послідовність фаз проекту, які необхідні для досягнення цілей проекту (від моменту визначення початку до закінчення проекту).



## Структура проекту

**Структура робіт проекту** – ієрархічна структура, яка поділяє роботу за проектом на групи, підгрупи, фази та ін. (називається WBS-структура проекту).

**WBS-структура** – це перелік робіт, що необхідно виконати в процесі реалізації поставленої цілі. Для її побудови необхідно провести декомпозицію робіт за допомогою декількох методів: за життєвим циклом або за змістом продукту. Для початку виділяють блоки робіт, які під час розробки WBS-структури необхідно деталізувати.

Необхідно відмітити появу **ризиків**. Причиною виникнення ризиків є невизначеність, що існує в кожному проекті. Ризики бувають:

- “відомі” – були визначені, оцінені, для них можливий варіант планування;
- “невідомі” – ризики, які не були ідентифіковані та не можуть бути прогнозованими.

## Автоматизовані системи управління ІТ-проектами

Основними функціональними можливостями автоматизованих систем управління ІТ-проектами є:

- ❑ засоби опису робіт проектів та зв'язків між ними;
- ❑ засоби інформаційного забезпечення;
- ❑ можливі витрати та контроль за виконанням проекту;
- ❑ графічні засоби представлення структури проекту.

Найбільш поширені автоматизовані системи управління ІТ-проектами – це *Microsoft Project, Agile, Scrum, Primavera Project Planner, Kanban, Time Line 6.5, Jira, Artemis Views, Spider Project, Open Plan, Basecamp, Trello, Slack, Asana, Worksection*.



:

- 1.
- 2.
- 3.
- 4.
- 5.
- 6.

,

,

,

–

:

- 1.
- 2.
- 3.
- 4.
- 5.
- 6.
- 7.

( )

–

:

.

–

:

- 1.
- 2.
- 3.
- 4.

– Agile.

– -

SMART

– Jira

:

**SMART :**

- S – specific, significant, stretching – , .  
 , .
- M – measurable, meaningful, motivational – , , .  
 , .
- A – attainable, agreed upon, achievable, acceptable, action-oriented – ,  
 , .  
 , / .
- R – realistic, relevant, reasonable, rewarding, result-oriented – ,  
 , .

，  
 .  
 – T – time-based, timely, tangible, trackable – , ,  
 . –  
 . , .

- ：
1. (roadmap).
  2. / .
  3. .
  4. .

- ：
1. 100% .
  2. .
  3. .
  4. .

- ：
1. : , , .
  2. .
  3. .
  4. .
  5. , .
  6. .
  7. , .
  8. : , .
  9. , .

- ：
1. .
  2. .
  3. : ,
  4. : .
  5. : , .
- , .

(.10):



. 10.



## Автоматизована система управління ІТ-пректом





## Приклади систем управління ІТ-проектами

**Slack** (з англ. "в'ялий, лінивий"). Основною складовою сервісу є чат (2013 рік створення), вважається корпоративним месенджером. Адміністратор групи може створювати різні канали, обмежуючи доступ однієї групи користувачів та делегуючи іншій. Це дозволяє уникнути інформаційної перенавантаження (напр., коли дизайнер змушений читати/слухати всі спірні питання розробників, пропускаючи важливі вимоги верстальника). Проект створювався як засіб комунікації та відповідає поставленим вимогам.

**Trello.** Це безкоштовний web-додаток (2011 рік створення) для управління проектами невеликих команд, сервіс розроблений Fog Creek Software. В основу покладено популярну парадигму управління *канбан* – вона полягає в тому, що за умови правильно заданих вихідних даних та вірному розподілу ресурсів – робота буде виконана вчасно. Розробники надають доступ до API, також доступні мобільні додатки на iOS та Android.

**Asana.** Основна одиниця сервісу – робочий стіл (віддалений роди "дошки" з Trello). Має можливість сортувати завдання за пріоритетом, присутня зручна система коментарів та зв'язку з колегами. Система надає можливість користуватись API, доступні додатки для iOS та Android. Схема підписки – Freemium, а за додатковий функціонал команді з 5 чоловік необхідно платити \$21.

## Приклади систем управління IT-проектами

**Basecamp.** Найвідоміша система (2004 рік створення), під час її розробки народився найвідоміший фреймворк Ruby on Rails. Розробники постійно вдосконалюють якість: представлення доступу до API, підтримка мобільних клієнтів, постійний випуск віджетів та плагінів для інтеграції в інші продукти.

**Worksection.** Вітчизняна розробка (2008 рік створення), задумувався як конкурент Basecamp, на даний момент ним і є, але лише на території СНД. Використовується модель Freemium (однак для волонтерських організацій надано безкоштовну можливість користуватись додатковими функціями). Безкоштовно надається 50 Мб місця на диску та 2 активних проекту. За \$19 команді надають доступ до всіх основних функцій, 5 Гб на диску та можливість вести 25 активних проектів.

## Slack

Переваги	Недоліки
<ul style="list-style-type: none"><li>✓ 5 Гб місця в "хмарі" на команду при безкоштовному використанні</li><li>✓ Зручний інтерфейс</li><li>✓ Інтеграція в більшість популярних сервісів розробки (як супутній продукт)</li><li>✓ Архів повідомлень доступний з будь-якого клієнта та в будь-який час</li></ul>	<ul style="list-style-type: none"><li>✓ Продукт занадто простий, щоб називатись системою управління проектами</li><li>✓ Комунікацію необхідно розділяти: частину залишити в основній системі управління та частину винести в Slack, що призводить до плутанини</li><li>✓ В безкоштовному архіві кількість повідомлень складає лише 10 000, а платна версія не має багато додаткових функцій</li></ul>
<p><b>ВИСНОВОК:</b> "Чайник для затишної кімнати гуртожитку".</p> <p>Ця система допоможе налагодити комунікацію в команді, однак на більше сподіватись не варто.</p> <p>Самостійною та повноцінною Slack назвати важко, але якщо мова йде про безкоштовне та просте рішення для невеликої команди, даний продукт буде чудовим варіантом.</p>	



# Trello

Переваги	Недоліки
<ul style="list-style-type: none"> <li>✓ Легка у використанні та доступна</li> <li>✓ Інтеграції у хмарні сховища для управління файлами</li> <li>✓ Зручні “дошки” (“що зробити”, “що робиться”, “що зроблено”) та чати</li> <li>✓ Сповіщення (повідомлення)</li> </ul>	<ul style="list-style-type: none"> <li>✓ Комплексний недолік – невеликий функціонал: “дошка” зручна лише для малих проектів, а для великих проектів функціонал не пристосовується</li> </ul>
<p><b>ВИСНОВОК:</b> “Мікрохвильова для молодої сімейної пари”.</p> <p>Використання цієї системи буде зручним, якщо необхідно організувати роботу, яка має бути виконана в обумовлений термін та не потребує особливої деталізації.</p> <p>Freemium-модель дозволяє не турбуватись щодо сплати непотрібних додаткових функцій для маленьких проектів, однак існує можливість “розширити” (оплатити) функціонал для роботи з великими проектами.</p>	

# Asana

Переваги	Недоліки
<ul style="list-style-type: none"> <li>✓ На високому рівні побудована функціональність, декілька “рівнів” постановки задачі</li> <li>✓ Гарна деталізація</li> <li>✓ Зручний інтерфейс зробить продукт схожим на повноцінний десктопний додаток</li> <li>✓ Відсутня необхідність в комунікації за межами сервісу</li> <li>✓ Система підходить як для планування серйозних задач, так і для реалізації простого плану</li> </ul>	<ul style="list-style-type: none"> <li>✓ Відсутність локалізованої версії продукту</li> <li>✓ У порівнянні з сервісом Trello, Asana надає дуже мало свободи для побудови “дошок-списків” за рахунок малої регламентації</li> </ul>
<p><b>ВИСНОВОК:</b> “Потужний блендер на кухні відомого ресторану”.</p> <p>Сервіс знадобиться різношерстим командам і там де необхідно розділяти “робочі столи” різних груп користувачів.</p> <p>За рахунок зручних списків задач сервіс допомагає організувати роботу команди не виходячи за межі системи – “Командна робота без e-mail”.</p>	

# Basecamp

Переваги	Недоліки
<ul style="list-style-type: none"> <li>✓ Простота та розподіленість</li> <li>✓ Інтуїтивний інтерфейс</li> <li>✓ Інтеграції у відомі сервіси для розробки ПЗ</li> <li>✓ Доступна програма для викладачів та студентів</li> <li>✓ Існує можливість створювати власні доповнення</li> </ul>	<ul style="list-style-type: none"> <li>✓ Суворо обмежений набір послуг</li> <li>✓ Труднощі в кастомизації: веб-версія слабко перелаштовується під необхідні вимоги</li> <li>✓ Громіздкість: не зважаючи на простоту використання, продукт має велику кількість можливостей, які будуть зайвими в малих проектах</li> </ul>
<p><b>ВИСНОВОК:</b> <i>“Мультиварка для досвідченої команди кухарів”.</i></p> <p>За використання цієї системи необхідно сплачувати щомісячний платіж.</p>	

# Worksection

Переваги	Недоліки
<ul style="list-style-type: none"> <li>✓ Необмежена кількість користувачів не залежно від виду підписки</li> <li>✓ Синхронізація з відомими сервісами (напр. календар Google)</li> <li>✓ Можливість зберігати проекти на своєму сайті, використовувати SSL-шифрування</li> <li>✓ Зручна система обмеження доступу</li> <li>✓ Широка сфера застосування за рахунок величезного функціоналу</li> </ul>	<ul style="list-style-type: none"> <li>✓ Дуже малий об'єм наданого місця в безкоштовній версії</li> <li>✓ Не дуже зручна система інформування: керівнику необхідно моніторити статус виконання задачі</li> <li>✓ Важко задати чіткий час виконання задачі: цей пункт необхідно оговорювати окремо</li> </ul>
<p><b>ВИСНОВОК:</b> <i>“Потужна вітчизняна м'ясорубка для великої сім'ї”.</i></p> <p>Сервіс є адаптивним, дозволяє використовувати продукт на будь-яких пристроях, що компенсує відсутність мобільних додатків.</p>	





Microsoft Office,

Project Guide,

,

,

.

,

,

-

,

.

.

.

,

,

,

,

.

(

,

),

,

,

.

«

,

...».

,

.

,

.

,

.

.

.

,

.

Visual Reports

Microsoft Office Excel

Microsoft Office Visio Professional

,

,

,

Project.

.

,

.

.

:

;

;

.





## ВИЗНАЧЕННЯ

**Технічне завдання (ТЗ)** - документ, оформлений в установленому порядку, визначає цілі створення програми, вимоги до програми і основні вихідні дані, необхідні для її розробки, а також план-графік створення програми.

**ГОСТ 34.602-89** є основним документом, що визначає вимоги і порядок створення (розвитку або модернізації) інформаційної (автоматизованої) системи, відповідно до якого проводиться її розробка і приймання при введенні в дію.

**ГОСТ 19.201-78** регламентує розробку програм і складання технічного завдання.

**ГОСТ 2.114-95** регламентує розробку системи конструкторської документації, технічні умови.

# СКЛАДОВІ ТЕХНІЧНОГО ЗАВДАННЯ

## Найменування розробки та область застосування

- умовне позначення об'єкту, шифр теми або номер договору;
- перелік документів, на підставі яких створюється проект, ким і коли затверджені ці документи;
- планові терміни початку та закінчення робіт із створення об'єкту.

## Підстава для розробки та назва проектної організації

- найменування підприємств розробника і замовника системи та їхні реквізити;
- перелік юридичних та фінансових документів, на підставі яких створюється система, ким і коли затверджені ці документи;
- відомості про джерела та порядок фінансування робіт.

## Мета розробки

## Джерела розробки

- техніко-економічне обґрунтування, звіти про закінчені науково-дослідні роботи, інформаційні посилання на вітчизняні й зарубіжні аналоги та ін.

## Технічні вимоги

- склад об'єкту та вимоги до його конструктивного виконання;
- вимоги до надійності, технологічності, рівня уніфікації та стандартизації, патентної чистоти та ін.

# СКЛАДОВІ ТЕХНІЧНОГО ЗАВДАННЯ

## Економічні показники

- гранична ціна, економічний ефект, термін окупності витрат на розробку і освоєння об'єкту, допустима річна потреба в об'єкті проектування.

## Порядок контролю і приймання об'єкту

- види, склад, обсяг і методи випробувань системи та її складових частин;
- загальні вимоги до приймання робіт (продукції) за стадіями, порядок узгодження і затвердження приймальної документації;
- статус приймальної комісії.

До технічного завдання можуть входити і такі розділи:

- ☐ **Допущення і обмеження.** Як правило, цей розділ заповнюється Виконавцем, у ньому перераховують наступні припущення та обмеження: функціональність, що виходить за рамки проекту, завдання, що виходять за рамки проекту, технічні обмеження, залежно від зовнішніх умов, які можуть вплинути на прийняті зобов'язання.
- ☐ **Ризики.** Це фактори, які можуть вплинути на вартість і терміни виконання робіт.

# КОМУ І ДЛЯ ЧОГО ПОТРІБНО ТЗ

<b>Обом сторонам:</b>	
<ul style="list-style-type: none"> <li>✓ представити готовий проект до початку роботи;</li> <li>✓ послідовно виконати перевірку готового продукту;</li> <li>✓ зменшити кількість помилок, пов'язаних зі зміною вимог внаслідок їхньої неповноти або хибності.</li> </ul>	
<b>Замовнику:</b>	<b>Виконавцю:</b>
<ul style="list-style-type: none"> <li><input type="checkbox"/> усвідомити, що саме йому потрібно, чітко це сформулювати;</li> <li><input type="checkbox"/> вимагати від виконавця відповідності продукту всім обумовленим та затвердженим пунктам ТЗ.</li> </ul>	<ul style="list-style-type: none"> <li><input type="checkbox"/> зрозуміти суть поставленого завдання;</li> <li><input type="checkbox"/> планувати виконання проекту в деталях і працювати за наміченим планом;</li> <li><input type="checkbox"/> відмовитися від виконання робіт, не зазначених у ТЗ.</li> </ul>

## ЦІЛІ ТЕХНІЧНОГО ЗАВДАННЯ



## СПИСОК ВИМОГ, ЯКІ НЕОБХІДНО ВИКЛАСТИ В ТЕХНІЧНОМУ ЗАВДАННІ:

- × до набору виконуваних програмою функцій;
- × до організації вхідних та вихідних даних;
- × до швидкодії та надійності функціонування,
- × до тривалості відновлення при відмовах;
- × до відмов у зв'язку з некоректними діями користувача;
- × до видів обслуговування, захисту та інформаційної безпеки;
- × до числа і кваліфікації персоналу, що взаємодіє з програмою;
- × до параметрів технічних засобів, на яких буде забезпечуватися нормальна працездатність програми;
- × до вихідних мов та кодів програмування, інформаційних структур і сторонніх програмних засобів;
- × до маркування та пакування, до умов транспортування і зберігання.



1. .
2. ,
3. .
4. .
5. .
6. .
7. - .
8. .
9. .
10. .
11. .
12. .
13. .
14. .
15. ,  
.
16. .
17. .
18. , , ,  
.
19. , .
20. .
21. .
22. .
23. ( ) .
24. .

,  
 ,  
 .  
 ,  
 :  
 -  
 ,  
 ,  
 -  
 .  
 ,  
 :

/	,
1	
2	
3	-
4	
5	
6	
7	
8	<i>Axure RP</i> -
9	

/	( , )
1	- . - . : . : 1.



2	<p>.</p> <p>:</p> <p>:</p> <p>2.</p>
3	<p>.</p> <p>:</p> <p>:</p> <p>3.</p>
4	<p>.</p> <p>:</p> <p>:</p> <p>4.</p>
5	<p>,</p> <p>:</p> <p>:</p> <p>5.</p>
6	<p>.</p> <p>:</p> <p>:</p> <p>6.</p>
7	<p>-</p> <p>:</p> <p>:</p> <p>7.</p>
8	<p>.</p> <p>:</p> <p>:</p> <p>8.</p>
9	<p>.</p>

	<p> : </p> <p> . </p> <p> : 9. </p>
10	<p> <u>1</u> </p> <p> : « , » </p> <p> : 1. </p>
11	<p> . </p> <p> - . </p> <p> : </p> <p> . </p> <p> : 10. </p>
12	<p> : . </p> <p> : - . </p> <p> : </p> <p> . </p> <p> : 10. </p>
13	<p> . </p>

.  
 ( ):  
 - ,  
 .  
 .  
 ( ):  
**1** , - .  
**2** , - .  
**3** - .  
**4** .  
**5** - .

( ) *1.* - (   
 1.1. . 1.2. . 1.3. ,  
 . 1.4. .);  
 2.  
 ( 2.1. . 2.2. 2.4.  
 . 2.3. - .  
 .).  
**1** - ,  
 ,  
 ,  
 .

1. . . . . – . . . . , 2004. – 666 .
2. . . . . : . . . . – . . . . , 2005. – 1(13). – . 5-19.
3. . . . . : ; [ . . . . . – . . . . : « »], 2003. – 528 .
4. . . . . – . . . . , 2006. – 432 .
5. - . . . . . – . . . . : « » , 2001. – 640 .
6. . . . . . . . . . . – . . . . « » , 2002. – 240 .
7. . . . . . . . . . : . . . . , 2010. – 432 .
8. . Scrum: . . . . , 2011. — . 576.
9. . . . . , . . . . . . . . . . , 2004. – 752 .
10. . Scrum XP: . – C4Media, 2007. – . 140.
11. . Scrum: . . . . , 2011. – . 576.
12. . Scrum. . . . . , 2016. – 288 .
13. . Scrum: . . . . , 2016. – . 544.
14. . . . . //
15. *Lester Albert*. Project Planning and Control. Fourth Edition. – Oxford OX2 8DP: Elsevier Butterworth-Heinemann Linacre House, 2003. – 382 p.